Data-driven Parameterized Policies for Microgrid Control

Ioannis Boukas, Sélim El Mekki and Bertrand Cornélusse
*Department of Electrical Engineering and Computer Science
University of Liège, Liège, Belgium
Email: {ioannis.boukas, selim.elmekki, bertrand.cornelusse}@uliege.be

***Abstract*—If we account for data uncertainty, centralized microgrid control can be decomposed in four tasks: estimating the parameters of the microgrid devices, forecasting the consumption and the renewable production, operational planning to anticipate weather effects and human activities, and real-time control to adapt planned decision to the reality of the moment. As the devices of a microgrid deteriorate over their lifetime and microgrids are by definition small systems, it is of paramount importance to automate the meta-parameterization of these stages to maximize the microgrid efficiency and decrease its maintenance costs. This paper studies how reinforcement learning can be used to address this problem. As reinforcement learning makes use of microgrid operation data (or of simulated data before the microgrid is actually operated) to learn an operation policy, it inherently merges the four stages above, and can in theory adapt to some types of changes without having to perform manual tuning.**

***Index Terms*—Microgrid control, optimization, reinforcement learning, uncertainty.**

I. Introduction

Microgrids are small electrical networks composed of flexible consumption, distributed power generation (renewable and/or conventional) and storage devices. The operation of a microgrid is optimized in order to satisfy the demand while ensuring maximum reliability and power quality and to maximize the renewable energy harvested locally while minimizing the total system cost.

Centralized microgrid control is usually decomposed in four tasks: i) estimating the parameters of the microgrid devices (for instance the charge efficiency of a battery storage device as a function of the state of charge and temperature, or the actual capacity of a battery after a number of cycles), ii) forecasting the consumption and the renewable production, iii) operational planning to anticipate weather effects and human activities, and iv) real-time control to adapt planned decision to the reality of the moment. These tasks are preformed sequentially during the lifetime of a microgrid in order to achieve near optimal operation and to maximize the benefits arising from distributed generation.

The estimation of the system parameters is a critical task for the optimization of the microgrid operation. The most important parameters are the operation costs as well as the capacities of the different components and the battery efficiency [1]. This process is usually carried out using measured data and is very specific to each microgrid configuration. These parameters are then used in the simulation model where the system operation is modeled.

After the parameters' estimation, it is important for the efficient microgrid operation to incorporate in the decision making process all the sources of uncertainty. To this end, forecasting techniques are deployed for the stochastic production and consumption. Forecasts are collected several times before the physical delivery in order to improve the accuracy of the forecasted value in the light of new information. There is a variety of forecasting techniques in the literature ranging from fundamental models of consumption and renewable energy production [2] to statistical models using measured data [3].

Following the recent advances in artificial intelligence and the data availability, the forecasting of renewable energy sources (RES) is proposed using artificial neural networks (ANN) in [4], where a hybrid model combines ANNs together with a Clear Sky Solar Radiation Model (CSRM) for the PV forecasting. The resulting model is using weather forecasts and real hourly photovoltaic power data. Short-term load forecasting (STLF) for microgrids using artificial intelligence is proposed in [5]. A three-stage architecture where first a self-organizing map (SOM) is applied to the input. The outputs are clustered using k-means algorithm, and finally demand forecasting for each cluster is performed with an ANN.

Subsequently, the outputs of the forecasting models in combination with the system parameters are used to compute the optimal control actions that need to be taken. The optimization of the control actions can be performed using the simulation model of the microgrid. However, the nonlinearities introduced by the system components make this problem hard to solve and without any optimality guarantees. Therefore, it is common in the literature to use a mixed integer linear approximation of the system model that can be solved easy with modern techniques and to optimality. A rolling horizon strategy is then usually adopted where the optimization is performed with some predefined look-ahead period [6]. Alternatively, a model predictive control (MPC) strategy is used for achieving economic efficiency in microgrid operation management [1]. An MPC policy is a feedback control law meant to compensate for the realization of uncertainty.

Given the data availability, the two preceding tasks, namely

Submitted to the 21st Power Systems Computation Conference (PSCC 2020).

forecasting and optimization, can be merged into one task and a control action can be derived directly from the data observed. To this end, reinforcement learning can be leveraged as a methodology to deal with the uncertainty and the non-linearities of the system components. The benefit from this approach is twofold, i) the nonlinearities of the simulator are accounted for in the optimization process and ii) the training of parameters is performed in the direction of an objective function that corresponds to the system cost instead of the mean squared error that is in the case of forecasting methods.

A modeling framework for the control of the storage device in the context of an interconnected microgrid is presented in [7]. Optimal Q-values are computed using the Q-learning method. In this setting the state and the action spaces are discretized in order to reduce the computational complexity and the results show increased utilization of the RES production compared to optimization methods.

Following the recent advancements in the field of Deep Reinforcement Learning (DRL), a Deep Q-learning approach is proposed in [8] for the control of seasonal storage in an isolated microgrid. In this framework, a specific deep learning structure is presented in order to extract information from the past RES production and consumption as well as the available forecasts. Despite the highly dimensional continuous state space, a control policy that is able to utilize optimally the storage is obtained.

In this paper, we present an open-source reinforcement framework for the modeling of an off-grid microgrid for rural electrification. Moreover, we formulate the control problem of an isolated microgrid as a Markov Decision Process (MDP). Due to the high-dimension continuous action space we define a set of discrete meta-actions in a similar way to [9]. We apply state of the art techniques for solving both the continuous and the discrete case. We investigate how reinforcement learning can utilize simulated data in order to learn an operation policy that minimizes the total system cost. Two benchmarks are selected for the performance evaluation of the obtained policy. A rule-based control that takes decisions in a myopic manner based only on current information and an optimization-based controller with look-ahead is applied are considered for comparison purposes.

This paper is organized as follows. Section II describes the simulation environment used for applying the methods presented in Section IV to the problem stated in Section III. Section V describes the case study and results. Section VI concludes and provides avenues for future research.

II. Simulator, models and reference controllers

To apply the proposed RL methods, it is of paramount importance to have a realistic simulator of the target environment for learning good policies before applying them to the actual system. This section details how our simulator, available as open source[1], is implemented in OpenAI gym [10]. The gym framework standardizes the way an environment is implemented, which allows applying existing algorithm implementations to a problem. The environment is essentially made of two methods: one method encodes the system dynamics, i.e. how the state evolves as a function of time, actions and random variables; the other method computes the reward associated to each state transition. The simulator is responsible for the detailed modeling and control of the microgrid components. It receives as input the microgrid configuration (components size and parameters, time series representing exogenous information, and simulation parameters) and simulates the operation for a predefined simulation horizon T.

[1] Available at https://github.com/bcornelusse/microgridRLsimulator.

A. Dynamics

The simulated system is composed of several consumption, storage and generation devices. For simplicity, we omit the device index in this section.

1) Consumption: The consumption of the isolated microgrid $C^{\text{non flexible}}$ is considered to be non-flexible, meaning that there is a high cost associated to the energy non-served. The consumption $C_t^{\text{non flexible}}$ at each time-step t of the simulation is assumed to be a stochastic variable sampled from some distribution P_C according to:

$$C_t^{\text{non flexible}} \sim P^C(\cdot). \tag{1}$$

In this paper, it is represented by real data gathered from an off-grid microgrid.

2) Storage model: The modeling of the storage system can become quite complex and highly-nonlinear depending of the degree of accuracy required by each specific application. In this paper, we use a linear "tank" model for the simulation of the battery since we assume that the simulation time-step size Δt is large enough (1 hour). The dynamics of a battery are given by

$$SoC_{t+1} = SoC_t + \Delta t \cdot (\eta^{\text{charge}} P_t^{\text{charge}} - \frac{P_t^{\text{discharge}}}{\eta^{\text{discharge}}}), \tag{2}$$

where SoC_t denotes the state of charge at each time step t, P^{charge} and $P^{\text{discharge}}$ correspond to the charging and discharging power respectively and η^{charge}, $\eta^{\text{discharge}}$ represent the charging and discharging efficiencies of the storage system. The charging (P^{charge}) and discharging ($P^{\text{discharge}}$) power of the battery are assumed to be limited by a maximum charging $\overline{P}$ and discharging $\underline{P}$ rate respectively. Accounting for the storage system degradation, we consider that the maximum capacity $\overline{S}$ of the storage system as well as the charging and discharging efficiencies (η^{charge}, $\eta^{\text{discharge}}$) are decreasing as a linear function of the number of cycles n_t that are performed at each time-step t. We have, $\forall t \in T$,

$$SoC_t, P_t^{\text{charge}}, P_t^{\text{discharge}} \geq 0 \tag{3}$$

$$P_t^{\text{charge}} \leq \overline{P} \tag{4}$$

$$P_t^{\text{discharge}} \leq \underline{P}, \tag{5}$$

$$SoC_t \leq \overline{S} \tag{6}$$

3) Steerable generator model: Steerable generation is considered any type of conventional fossil-fuel based generation that can be dispatched at any time-step t. When a generator is activated, it is assumed to operate at the output level P_t^{steer} that is ranging between the minimum stable generation $\underline{P^{\text{steer}}}$ and the maximum capacity $\overline{P^{\text{steer}}}$ such that

$$\underline{P^{\text{steer}}} \leq P_t^{\text{steer}} \leq \overline{P^{\text{steer}}}. \tag{7}$$

The fuel consumption F_t related to the operation of the generator at time t is a linear function of the power output P_t^{steer} curve with parameters F_1, F_2 given by the manufacturer.

$$F_t = F_1 + F_2 \cdot P_t^{\text{steer}}. \tag{8}$$

The fuel cost c_t^{fuel} accounting for the fuel price π^{steer} is then given by:

$$c_t^{\text{fuel}} = F_t \cdot \pi^{\text{fuel}}. \tag{9}$$

4) Non-steerable generators model: The level of non-steerable generation from renewable resources such as wind or solar is denoted by $P^{\text{non steer}}$. Similar to the non-flexible load case it is assumed that $P_t^{\text{non steer}}$ at time-step t is sampled from a probability distribution $P_{P^{\text{non steer}}}$ according to:

$$P_t^{\text{non steer}} \sim P_{P^{\text{non steer}}}(\cdot). \tag{10}$$

In this paper, this is represented by real data gathered from an off-grid microgrid.

5) Power balance: At each time-step t in the simulation horizon we can compute the power balance between the injections and the off-takes. The residual power P_t^{RES} resulting from the mismatch between production and consumption is curtailed P_t^{curt} if its positive and shed C_t^{shed} if it is negative. We can formally define the power balance as:

$$\begin{aligned} P_t^{\text{non steer}} + P_t^{\text{steer}} + P_t^{\text{discharge}} - P_t^{\text{charge}} - C_t^{\text{non flexible}} \\ = P_t^{\text{curt}} - C_t^{\text{shed}}, \end{aligned} \tag{11}$$

with $P_t^{\text{curt}}, C_t^{\text{shed}} \geq 0$. The costs arising from the curtailment of generation or the shedding of non-flexible loads are given by:

$$c_t^{\text{curt}} = P_t^{\text{curt}} \cdot \pi^{\text{curt}} \tag{12}$$

$$c_t^{\text{shed}} = C_t^{\text{shed}} \cdot \pi^{\text{shed}} \tag{13}$$

B. Rule-based controller

The Rule-based controller is a simple myopic controller that implements a set of decision rules to determine the control actions that need to be taken at each time-step t. It requires only data regarding the present condition of the microgrid. It serves as a baseline for comparison purposes, and as a post-processing for our proposed methods as described in Section IV. The logic that is implemented is the following:

1) First, the difference between the current total RES production and non-flexible demand is computed:

$$P_t^{RES} = P_t^{\text{non steer}} - C_t^{\text{non flexible}}$$

Algorithm 1 Power dispatch.

Inputs: P_t^{RES} , $y_t = (b.Status, \forall b \in \mathcal{B})$
if $P_t^{RES} \geq 0$ **then**
  for all $b \in \mathcal{B}$ **do**
    if $b.Status =$ "C" **then**
      $P_t^{\text{charge}} = \min(P^{RES}, \overline{P})$
    end if
    $P_t^{RES} \leftarrow P_t^{RES} - P_t^{\text{charge}}$
  end for
else
  for all $b \in \mathcal{B}$ **do**
    if $b.Status =$ "D" **then**
      $P_t^{\text{discharge}} = \min(-P^{RES}, \underline{P})$
    end if
    $P_t^{RES} \leftarrow P_t^{RES} + P_t^{\text{discharge}}$
    if $P_t^{RES} \leq 0$ **then**
      $P_t^{\text{steer}} = P_t^{RES}$
    end if
  end for
end if

2) If P_t^{RES} is positive, the status of every battery $b \in \mathcal{B}$ is set to charge ("C") and the vector y_t is formed as:

$$y_t = (\text{"}C\text{"}, \forall b \in \mathcal{B})$$

3) If P_t^{RES} is negative, the status of every battery $b \in \mathcal{B}$ is set to discharge ("D") and the vector y_t is formed as:

$$y_t = (\text{"}D\text{"}, \forall b \in \mathcal{B})$$

4) When the vector y_t is fixed, the residual generation is dispatched over devices as presented in Algorithm 1, and the decision variables related to the storage devices ($P_{b,t}^{\text{discharge}}, P_{b,t}^{\text{charge}}, \forall b \in \mathcal{B}$) and the generators ($P_{g,t}^{\text{steer}}, \forall g \in \mathcal{G}$) are determined.

C. Optimization-based controller

The optimization-based controller serves as a baseline for comparison to our proposed methods. This controller receives as input all the parameters available such as the renewable production and the electrical demand forecasts, state of charge of the storage systems, components parameters, etc., and solves an optimization problem in receding horizon. The objective function to minimize aggregates curtailment, shedding and fuel costs (the π parameters denote unit costs):

$$\begin{aligned} \min \Delta_t \sum_t \Bigg(\sum_{g \in \mathcal{P}^{\text{non steer}}} \pi_g^{\text{curt}} P_{g,t}^{\text{curt}} + \sum_{d \in \mathcal{P}^{\text{non flexible}}} \pi_d^{\text{shed}} C_{d,t}^{\text{shed}} \\ + \sum_{g \in \mathcal{G}} \pi_g^{\text{fuel}} \left(F_{g,1} + F_{g,2} P_{g,t}^{\text{steer}} \right) \Bigg). \end{aligned} \tag{14}$$

The energy balance constraint is, $\forall t \in \mathcal{T}$:

$$\begin{aligned} \sum_{g \in \mathcal{G}} P_{g,t}^{\text{steer}} + \sum_{g \in \mathcal{P}^{\text{non steer}}} (P_{g,t}^{\text{non steer}} - P_{g,t}^{\text{curt}}) + \sum_{b \in \mathcal{B}} P_{b,t}^{\text{discharge}} \\ = \sum_{b \in \mathcal{B}} P_{b,t}^{\text{charge}} + \sum_{d \in \mathcal{P}^{\text{non flexible}}} (C_{d,t}^{\text{sheddable}} - C_{d,t}^{\text{shed}}) \end{aligned} \tag{15}$$

The non-negative variables $P_{b,t}^{\text{charge}}$ and $P_{b,t}^{\text{discharge}}$ are bounded above by the maximum charging $\overline{P}_b$ and discharging $\underline{P}_b$ power of battery b, respectively. For each generator g, the fraction of the non-steerable power generation $P_{g,t}^{\text{non steer}}$ that is curtailed and the fraction of steerable generation $P_{g,t}^{\text{steer}}$ that is activated are represented by $P_{g,t}^{\text{curt}}$ and $P_{g,t}^{\text{steer}}$, respectively. Binary variables $k_{g,t}$ are added to the model for the minimum operating point of the steerable generators, $\forall t \in \mathcal{T}$:

$$k_{g,t}\underline{P_g^{\text{steer}}} \leq P_{g,t}^{\text{steer}} \leq k_{g,t}P_g^{\text{steer}} \tag{16}$$

The transition law of the state of charge s of each battery b is modelled as in (9). This problem is thus a mixed-integer linear program.

III. Problem Statement

This section reformulates the discrete-time dynamic system described in the previous section as a Markov Decision process. At each time-step t, the state variable $s_t = \left((SoC_{b,t}, \forall b \in \mathcal{B}), P_t^{\text{curt}}, C_t^{\text{shed}}\right) \in S$ contains all the relevant information for the optimization of the system. The control $a_t = \left((P_{b,t}^{\text{discharge}}, P_{b,t}^{\text{charge}}, \forall b \in \mathcal{B}), (P_{g,t}^{\text{steer}}, \forall g \in \mathcal{G})\right) \in A$ applied at each time-step t contains the charging/discharging decisions for the storage systems and the generation level of the steerable generators. The exogenous variable $\omega_t = \left(\hat{P_{t,g}}^{res}, \hat{L}_{t,l}\right) \in \Omega$ corresponds to a random disturbance that is sampled from a probability distribution $\omega_t \sim P_\omega(\cdot)$. At each time-step t the system performs transitions based on the dynamics described in Section II-A according to:

$$s_{t+1} = f\left(s_t, a_t, \omega_t\right), \qquad t = 0, ..., T. \tag{17}$$

Each transition generates a cost c_t according to the cost function $c(s_t, a_t) \in \mathbb{R}$. In our problem, the cost function is:

$$c_t = c(s_t, a_t) = c_f + c_{curt} + c_{sh}. \tag{18}$$

Given an initial state s_0 and a sequence of actions $(a_0, ...a_{T-1})$, the total discounted cost at the end of the control horizon is computed as:

$$J^{(a_0,...a_{T-1})}(s_0) = \sum_{t=0}^{T-1} \gamma^t c(s_t, a_t), \tag{19}$$

where $0 < \gamma < 1$ is a discount factor. A closed-loop control policy $\pi = \{\pi_0, ..., \pi_{T-1}\} \in \Pi$, from the set of admissible policies Π is a rule for selecting a control action at each time-step t according to:

$$a_t = \pi_t(s_t) \in A, \qquad t = 0, ..., T. \tag{20}$$

The total discounted cost associated to a policy $\pi \in \Pi$ is given by:

$$J^\pi(s_0) = \sum_{t=0}^{T-1} \gamma^t c(s_t, \pi(s_t)). \tag{21}$$

An optimal policy $\pi^* \in \Pi$ is a policy that, for any initial state (s_0), yields the actions that minimize the total discounted cost J^*, such that:

$$J^* = \min_\pi J^\pi(s_0), \tag{22}$$

$$\pi^* = arg\min_\pi J^\pi(s_0). \tag{23}$$

Additionally the state-action value function $Q_t(s_t, a_t)$ associated to an optimal policy π^* is used to characterize the quality of taking action a_t at state s_t and then acting optimally and is defined as:

$$Q_t(s_t, a_t) = c(s_t, a_t) + \gamma \min_{a_{t+1}} Q_{t+1}(s_{t+1}, a_{t+1}). \tag{24}$$

The optimal action at each time-step t can be obtained using the optimal Q-value as:

$$\pi_t^*(s_t) = arg\min_{a_t} Q_t(s_t, a_t), t = 0, ..., T. \tag{25}$$

IV. Methodology

To solve the problem described in Section III, we first assume that the control horizon T is large enough so that a stationary policy $\pi = \{\pi(s_t), ..., \pi(s_t)\}$ is optimal. This assumption is necessary in order to use the two main classes of methods for solving infinite-horizon problems namely *Value Iteration* and *Policy Iteration*. The former provides a methodology for the computation of the optimal Q values so that equation (25) can be used to infer an optimal policy. The latter starts with an initial policy function that is updated sequentially using estimates of the cost to go J^π.

Due to the continuous and high dimensional nature of the state and the action spaces of the problem, these methods cannot be applied in their exact form. However, recent developments in the field of reinforcement learning have made possible the design of approximate optimal policies using function approximation techniques. In this paper, we use the state of the art methodology from each of the two classes of algorithms namely the Double Deep-Q networks with prioritized experience replay (value-based) and the Proximal Policy Optimization (policy-based).

A. Meta-actions

In this section, we elaborate on the design of a small and discrete set of actions A' that maps to the original action space A. This step is necessary for the use of value-based algorithms, as the minimization problem defined in equation (25) is hard to solve. Each of the storage devices can be in a status $y_{b,t} \in Y$ of charging, discharging or idling where $Y = \{\text{“}C\text{”}, \text{“}D\text{”}, \text{“}I\text{”}\}$. We define a discrete action $a'_t \in A'$ that can take values from all the possible combinations of $y_{b,t}$ as:

$$a'_t = (y_{b,t}, \forall b \in \mathbb{B}) \in Y^{|B|} \tag{26}$$

Defining the action space in this way allows the use of the dispatch rule defined in algorithm 1 to obtain the control actions a_t.

B. Double Deep Q Networks with prioritized experience replay

This algorithm combines the Q-learning algorithm [11] with the use of deep neural networks for the representation of the optimal Q-function, referred to as deep q networks (DQN). Let $Q(s_t, a'_t; \theta)$ denote a parametric approximation of the state-action value function Q with parameters θ. After each transition a quadruple $(s_t, a'_t, c_t, s_{t+1})$ is collected and the parameters θ are updated according to:

$$\theta_{t+1} = \theta_t - \alpha \cdot (Y^Q - Q(s_t, a'_t; \theta)) \cdot \nabla Q(s_t, a'_t; \theta), \quad (27)$$

where $0 < \alpha < 1$ is a step size and Y^Q is the update target defined as:

$$Y^Q \equiv c_t + \gamma \min_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}; \theta). \quad (28)$$

The DQN algorithm as proposed in [12] additionally makes use of a target network with parameters and experience replay. The target network, with parameters θ^-, is used for the generation of the targets Y^{DQN} as shown in equation (29). These targets are then used in equation (27) for the update of parameters θ of the main network. The parameters θ^- are only updated every n time-steps. With experience replay a batch of transitions is sampled from a buffer for updating the parameters θ. Both methods are shown to improve dramatically stability issues during the training process.

$$Y^{DQN} \equiv c_t + \gamma \min_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}; \theta^-). \quad (29)$$

The double Q-learning algorithm is an extension of DQN proposed in [13] to address issues of under-overestimation of the Q function. The target Y^{DDQN} used for the update in equation (27) is computed as:

$$Y^{DDQN} \equiv c_t + \gamma Q(s_{t+1}, \min_{a'_{t+1}} Q(s_{t+1}, a'_{t+1}; \theta); \theta^-). \quad (30)$$

In this approach the action is selected according to the main weights θ, while the second set of weights θ^- is used to fairly evaluate the value of the policy. Finally, prioritized experience replay is used instead of sampling the transitions uniformly as proposed in [14]. The motivation originates from the fact that important transitions should be visited more frequently.

C. Proximal policy optimization

Proximal policy optimization (PPO) [15] belongs to the family of policy gradient methods and can be used with both discrete and continuous action spaces. In the vanilla actor-critic method [16] a parametrized stochastic policy function $\pi(a_t|s_t; \theta)$ with parameters θ is directly optimized towards the objective defined in equation (23). After the collection of N full trajectories $\tau = (s_{0,i}, a_{0,i}, c_{0,i}, s_{t+1,i}, ..., s_{T,i})$ a gradient step is performed for the update of the parameters θ as:

$$\theta_{t+1} = \theta_t - \alpha \nabla J^\pi, \quad (31)$$

with

$$\nabla J^\pi = \mathbb{E}_{\tau_1,...,\tau_N} \left\{ \sum_{t=0}^{T} \nabla \log \pi(a_{t,k}|s_{t,k}; \theta) \hat{A}^\pi(s_{t,k}, a_{t,k}) \right\}, \quad (32)$$

where $\hat{A}(s_t, a_t)$ is an estimator of the advantage function. The advantage function represents how much better than average an action a_t is and is commonly represented as:

$$\hat{A}^\pi(s_{t,k}, a_{t,k}) = c(s_{t,k}, a_{t,k}) + \gamma \hat{J}^\pi_\phi(s_{t+1,k}) - \hat{J}^\pi_\phi(s_{t+1,k}), \quad (33)$$

It is empirically shown that the vanilla actor-critic method often leads to destructively large policy updates. To this end, PPO uses importance sampling and a clipping of the optimization objective in order to constrain the new policy in case the update is too large. Let $r(\theta)$ denote the probability ratio,

$$r(\theta) = \frac{\pi(a_{t,k}|s_{t,k}; \theta_{new})}{\pi(a_t|s_t; \theta_{old})}.$$

The clipped objective proposed in [15] can be written as:

$$J^{Clip} = \mathbb{E}\left\{ \max(r(\theta)\hat{A}^\pi, clip(r(\theta), 1-\epsilon, 1+\epsilon)) \right\} \quad (34)$$

The optimal policy is derived by performing multiple steps of stochastic gradient descent on this objective. While standard policy gradient methods perform one gradient update per data sample, the PPO algorithm enables multiple epochs of mini-batch updates resulting in better sample efficiency.

V. Case study

The evaluation of the developed methodology is performed using empirical data measured by the off-grid micro-grid system of the village "El Espino" (-19.188, -63.560), in Bolivia, installed in September 2015 and composed of 60kW of PV panels, 464kWh of battery storage and a 58 kW generator. The system serves a community of 128 households, a hospital and a school, as well as the public lighting service. A comprehensive description of the system and of the data is available in [17]. Aggregate electric load data is available as an indirect measure, i.e. as the sum of direct measurements retrieved from PV arrays, Gen-Set and batteries by means of smart meters. The parameters used for this specific microgrid configuration are given in Table I. It is important to note that the minimum stable generation level is rather high due to local regulations: the very low diesel price is due to state subsidy, but the state restricts the operation of all the conventional generators below 80% of their nominal capacity in order to operate close to the optimal efficiency. This restriction imposes a large discontinuity in the microgrid operation and is further discussed in the results section.

The following protocol was carried out for the training and the evaluation of the proposed algorithms. For the training process we adopted a rolling strategy as it would happen in practice. More specifically, the two policies were trained in the the first two months of 2016 and were tested in the third month. Then the third month became part of the training set

TABLE I: Input parameters.

$\overline{S}$	464	kWh
$\overline{P}$, $\underline{P}$	60	kW
η^{charge}, $\eta^{\text{discharge}}$	95 %	
π^{fuel}	1	€/kWh
π^{curt}	10.5	€/kWh
$\pi^{\text{lost load}}$	100	€/kWh
Δ_t	1	h
$\overline{P^{\text{non steer}}}$	60	kW
$\overline{P^{\text{steer}}}$	58	kW
$\underline{P^{\text{steer}}}$	46.4	kW

and after a second round of training the new policy was tested on the fourth month. This process was carried out until the dataset was exhausted. Finally we ran a separate experiment for which the training set considered was the full year of 2016 and we selected as a test set the first 7 months of 2017.

The performance of the algorithms is compared against the two benchmarks described in Section II. Three variances of the optimization controller were considered for comparison purposes. First, an optimization controller with perfect knowledge and 12 periods of look-ahead is considered in order to obtain a good approximation for the lower bound of the control problem. Second, an optimization controller with 12 periods of look-ahead and additional noise around the exact value of the stochastic variables. Third, a myopic optimization controller with no period of look-ahead was considered.

The results of the described protocol are presented in Figure 1. The total cost of each strategy for each testing period is shown, so that a comparison can be drawn. The PPO algorithm was significantly under-performing at the time of the experiment and therefore was omitted. Initially, we observe that in many cases the DQN policy is performing very close to the optimization controllers. Especially in the case of the April 2017 the DQN policy manages to obtain lower costs than the noisy forecast and the no look-ahead optimization controllers. The DQN policy is also found to perform better than the rule based controller in most cases. This is due to the fact that the DQN algorithm manages to anticipate periods of high energy curtailment or load shedding and manages to utilize the storage device accordingly.

Finally, in the last case where the performance of the strategies is evaluated in a longer period the results show that the DQN policy does not manage to outperform the optimization-based benchmarks. This is mainly due to the limitation introduced by the design of the meta-actions. In that respect a set of more elaborate meta-actions could be used to reduce further the cost.

VI. Conclusion

In this paper, we investigate whether data driven approaches can be used effectively for the control of a microgrid. First, an open-source reinforcement framework for the modeling of an off-grid microgrid for rural electrification is presented. The control problem of an isolated microgrid is casted as a Markov Decision Process (MDP). We deploy state of the art techniques for optimizing the operation of the system. We investigate how data coming from simulation can be used to learn an operation policy that minimizes the total system cost.

An experimental protocol was designed in order to train and test the proposed algorithms. Two benchmarks are selected for the performance evaluation of the obtained policy. A rule-based control that takes decisions in a myopic manner based only on current information and an optimization-based controller with look-ahead is applied are considered for comparison purposes.

The results indicate that the DQN policy was able to outperform in most test cases the myopic rule-based controller. Moreover, the DQN policy demonstrated results similar to the optimization controller with forecast. Further research should be directed towards the design of more elaborate meta-actions that manage to map better into the original action space.

VII. Acknowledgments

This research is carried out in the framework of the Dynamically Evolving Long-Term Autonomy (DELTA) project. DELTA is a European research project funded under the CHIST-ERA scheme (http://www.chistera.eu/). The authors would like to thank Sergio Balderrama for the provision of measured data from the ”El Espino” microgrid in Bolivia.

References

[1] A. Parisio, E. Rikos, G. Tzamalis, and L. Glielmo, “Use of model predictive control for experimental microgrid optimization,” *Applied Energy*, vol. 115, pp. 37–46, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2013.10.027

[2] A. Dolara, S. Leva, and G. Manzolini, “Comparison of different physical models for PV power output prediction,” *Solar Energy*, vol. 119, pp. 83–99, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.solener.2015.06.017

[3] F. Lombardi, S. Balderrama, S. Quoilin, and E. Colombo, “Generating high-resolution multi-energy load profiles for remote areas with an open-source stochastic model,” *Energy*, vol. 177, pp. 433–444, 2019. [Online]. Available: https://doi.org/10.1016/j.energy.2019.04.097

[4] S. Leva, M. Mussettav, A. Nespoli, and E. Ogliari, “PV power forecasting improvement by means of a selective ensemble approach,” pp. 1–5, 2019.

[5] L. Hernández, C. Baladrón, J. M. Aguiar, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, “Artificial neural networks for short-term load forecasting in microgrids environment,” *Energy*, vol. 75, pp. 252–264, 2014.

[6] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, and D. Saez, “A microgrid energy management system based on the rolling horizon strategy,” *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 996–1006, 2013.

[7] E. Kuznetsova, Y. F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell, “Reinforcement learning for microgrid energy management,” *Energy*, vol. 59, pp. 133–146, sep 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544213004817

[8] V. François-Lavet, D. Taralla, D. Ernst, and R. Fonteneau, “Deep reinforcement learning solutions for energy microgrids management,” in *European Workshop on Reinforcement Learning (EWRL 2016)*, 2016.

[9] I. Boukas, D. Ernst, A. Papavasiliou, and B. Cornélusse, “Intra-day Bidding Strategies for Storage Devices Using Deep Reinforcement Learning,” in *15th International Conference on the European Energy Market*, vol. 14, 2018.

[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[11] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

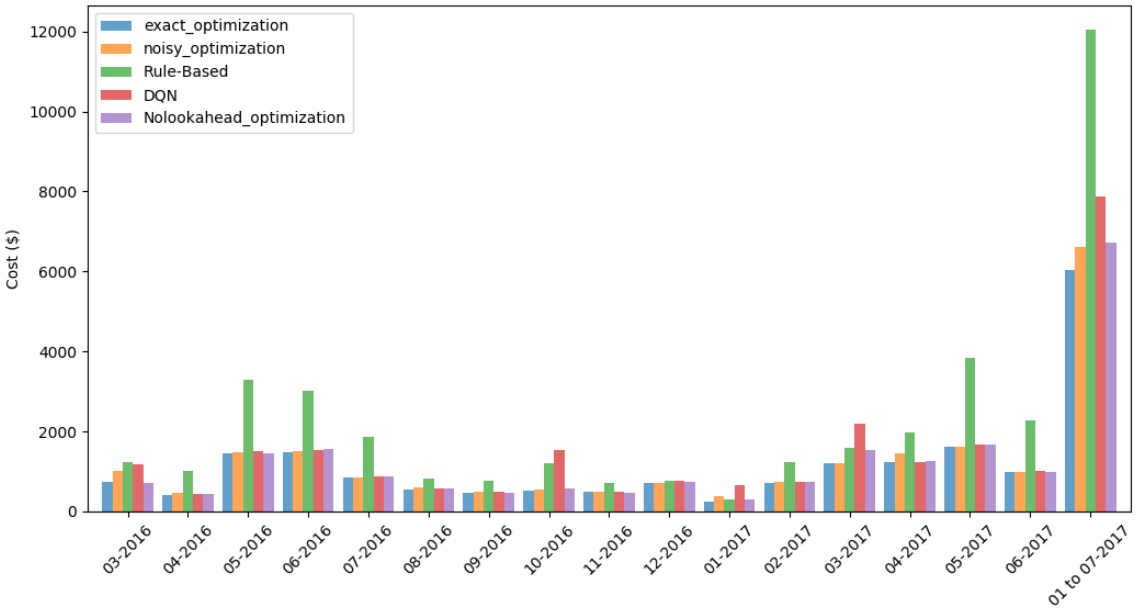

Fig. 1: Total cost for each control strategy in off-sample data.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[13] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100, 2016.

[14] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," pp. 1–21, 2015. [Online]. Available: http://arxiv.org/abs/1511.05952

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[16] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[17] S. Balderrama, F. Lombardi, F. Riva, W. Canedo, E. Colombo, and S. Quoilin, "A two-stage linear programming optimization framework for isolated hybrid microgrids in a rural context: The case study of the El Espino community," *Energy*, vol. 188, p. 116073, 2019. [Online]. Available: https://doi.org/10.1016/j.energy.2019.116073

NOTATION

Set and indices

- b battery index
- g conventional generator index
- t time period index
- $\mathcal{B}$ set of all batteries
- $\mathcal{G}$ set of all conventional generators
- $\mathcal{P}^{\text{non steer}}$ set of all non steerable generators
- $\mathcal{P}^{\text{non flexible}}$ set of all non flexible loads
- $\mathcal{A}$ action space
- $\mathcal{S}$ state space
- $\mathcal{T}$ set of periods in the control horizon

Parameters

- $\overline{P}$, $\underline{P}$ maximum charge and discharge rate (kW)
- $\overline{P^{\text{non steer}}}$ non steerable generation (kW)
- $\overline{P^{\text{steer}}}$ steerable generator capacity (kW)
- $\underline{P^{\text{steer}}}$ minimum steerable generation (kW)
- S^{init} initial state of charge (kWh)
- $\overline{S}$, $\underline{S}$ maximum and minimum battery capacity (kWh)
- Δ_t simulation and control period duration (h)
- η^{charge}, $\eta^{\text{discharge}}$ charge and discharge efficiency (%)
- π^{curt} curtailment cost (€/kWh)
- π^{fuel} fuel cost (€/kWh)
- $\pi^{\text{lost load}}$ lost load cost (€/kWh)

Variables

- a control actions vector
- P^{charge}, $P^{\text{discharge}}$ fraction of the maximum charging and discharging power [0,1]
- C^{shed}, load shed
- P^{curt} generation curtailed
- P^{steer} generation activated
- $P^{\text{non steer}}$ non-steerable generation
- $C^{\text{non flexible}}$ non-flexible load
- k binary variable
- c^{fuel} fuel cost (€)
- c^{curt} curtailment cost (€)
- c^{shed} lost load cost (€)
- SoC state of charge of battery (kWh)
- P^{charge} charged energy of battery (kWh)
- $P^{\text{discharge}}$ discharged energy of battery (kWh)